

A new calculator and why it is necessary *

Harold Thimbleby
Computing Science
Middlesex University
LONDON, N11 2NQ.
Email: harold@mdx.ac.uk

September 30, 1998

Abstract

Conventional calculators are badly designed: they suffer from bad computer science — they are unnecessarily difficult to use and buggy. I describe a solution, avoiding the problems caused by conventional calculators, one that is more powerful and arguably much easier to use. The solution has been implemented, and design issues are discussed.

This paper shows an interactive system that is declarative, with the advantages of clarity and power that entails. It frees people from working out *how* a calculation should be expressed to concentrating on *what* they want solved. An important contribution is to demonstrate the very serious problems users face when using conventional calculators, and hence what a freedom a declarative design brings.

1 Introduction

Pocket calculators have been around for decades, and it is easy to take their design for granted. (Even powerful computer systems simulate desktop calculators because of their familiarity and presumed ease of use.) However, conventional calculators are in fact poorly conceived: calculators made by market leaders are egregious, buggy and, in many cases, bizarre. This unfortunate state of affairs may be blamed on the arbitrariness permitted by imperative styles of design, which has enabled the uncontrolled accretion of incoherent, partial and undefined features, possibly motivated by naïve market forces. After substantiating these claims, this paper exhibits a new, clear calculator design that solves the identified problems and that has additional advantages. Thus I show the problems are technically avoidable. The new design is distinctively declarative; it demonstrates the supremacy of a declarative approach over the conventional *ad hoc* imperative approach. (The approach can be generalised to applications other than arithmetic but that is not my purpose here.)

My credit card agreement states that, “Interest is calculated on a daily basis at the rate of 1.585% per month.” If I have £276 credit for a week, how much interest do I pay? This question involves financial background knowledge, but consider the following direct arithmetic problem, which requires a similar level of mathematical skill to solve: “What power of 2 is 100?” and let us suppose we will use a conventional hand-held calculator to find the answer. The question is not in a form that is immediately acceptable, even in conventional symbols: $2^x = 100$. Although calculators typically have a key for raising to a power, it cannot be used for this sum because it assumes x is given, whereas here it is required to be calculated. Instead, the problem has to be compiled by the user to $x = \log 100 / \log 2$ and then to the particular button presses for a calculator, e.g., $\boxed{C} \boxed{1} \boxed{0} \boxed{0} \boxed{\log} \boxed{/} \boxed{2} \boxed{\log} \boxed{=}$. This sequence of key presses has very little resemblance to the original problem. (A similar effort is required for a reverse Polish calculator.)

More generally, the task of the user is to transform their problem into an appropriate sequence of commands that culminate in the calculator displaying the required answer. To do this correctly the semantics of the keys and their combinations must be well defined, ideally sensible and memorable. Thus pressing $\boxed{0}$ multiplies the display by 10 (when displaying less than 8 digits, when not following a

*This work was supported by EPSRC Grant No. GR/J43110.

decimal point, when not in an error condition . . .). Furthermore, as users may make slips — pressing the wrong button, omitting a press, or pressing a button twice in succession by accident — the intermediate displays of the calculator should confirm progress and, where an error occurs, help the user make an efficient repair.

This much is obvious, yet no known calculator has implemented a coherent syntax and semantics for its buttons. (We will shortly substantiate this claim by examining some common calculators made by market leaders.)

In all the conventional calculators I discuss here, there is unnecessary and avoidable obfuscation; and more complex operations are even more obfuscated. Similar problems are present in video recorders, digital clocks . . . nuclear powerstations and aircraft cockpits [19]. One purpose of this paper is to suggest that this is fundamentally a technical issue, and that the technical problems are avoidable.

Weiner describes a software manufacturer's shock that one of their spreadsheet packages was used in surgery [22], where a mistake could cost a patient's life. Would the manufacturer have made their product to a higher standard if they had thought it might be used in a safety critical application? It is however clear that calculators are routinely used in many safety and mission critical applications. It would clearly be outrageous to dismiss the problems of calculators as inconsequential or simply the user's responsibility, particularly when the problems are avoidable at negligible cost.

My informal surveys show that few users know what the percent key should do, whether in principle or on their own calculator. They prefer to work without it! Thus people adapt their behaviour and expectations to make their use of calculators more reliable. This adaptation to avoid problems can be so entrenched that some people may not appreciate this paper.

People who recognise problems with calculators tend to resign themselves, and blame problems on their own (presumed) incompetence. Manufacturers conversely blame users for not reading manuals or for not being 'technology literate,' which *surely* they ought to be [20]. Calculators perhaps engender this: they are 'mathematical,' an area itself that makes many people despair! So an important distinction must be made: of course, some calculators provide features to do complex things, but complexity is not the same as obfuscation. We cannot blame users for not understanding obfuscated systems; indeed it appears, if we are to believe poor design is not intentional, that calculators as presently conceived are also beyond the competence of manufacturers.

We need a new approach.

Note. Specific makes and models are mentioned to enable details to be checked; several models are discussed to indicate that problems are not restricted to one make or to one model. All are common models, in the price range £7–£50 (1994 prices), with the exception of a 1970s slide rule. I do not have space to exhaust any model's features in the discussion, and I am not attempting a comparative or other review of specific models or of their features. Inevitably our discussion raises problems; thus a reference to a particular calculator should not necessarily be taken to represent a manufacturer's or calculator's overall standard or range of features.

2 Specific problems with conventional calculators

"Mastery of the concepts underlying calculators is just the first step down the road to computer literacy." [12]

All calculators have quirks; they do complicated things. But for any particular quirk there is usually some calculator that does a sensible thing, proof that the quirk is avoidable.

Take any calculator (even a simulation running on a workstation) and work out 4×-5 , which should be -20 . Most calculators I have tried this sum on give -1 , *without reporting an error*.

What you are supposed to do to get the right answer is press keys: $[\underline{4}][\underline{\times}][\underline{5}][\underline{+/-}]$, the last key being a 'change sign' key that makes the 5 into -5 . Many calculators do not have $[\underline{+/-}]$ (such as the Casio MS-70L, Aurora B30, Texet SL-22, and many others) so they *cannot* do this sort of sum at all — yet these calculators all have a minus sign, so they could do it in principle. An exception is the Casio *fx-P401*, which does get 4×-5 correct using its normal $[\underline{-}]$ minus sign key, but requires a special key $[\underline{(-)}]$ to solve -4×5 correctly. We will see below further problems with $[\underline{+/-}]$.

All might be harmless design variation, except manufacturers' claims suggest otherwise, as the following typical example from a market leader makes clear:

"Thanks to V.P.A.M. (Visually Perfect Algebraic Method) calculations have become a lot easier. It may sound confusing, but all it means is that the scientific calculators in the Casio VPAM range perform calculations exactly as you would write them or read them in a text book.

For example, using the FX115s illustrated here, you would enter the calculation $5 \times 2 + \sin 30 = 10.5$ exactly as you would write it." [6]

It is quibbling to mention that you cannot enter the 10.5 "exactly as you would write it"; however, more seriously, the *fx-115s* cannot do $5 \times 2 + \sin -30$, $5 + 2 \sin 30$, or many other formulæ that might be taken straight out of a school text book. Nor does the calculator report errors when it obtains wrong answers. The manufacturer's claims are inflated, and the unwary user is not protected.

Many calculators have memory restrictions: the Casio *fx-82LB* manual ominously warns you not to use memory calculations in "SD" mode (why not? what goes wrong?), though the Hewlett-Packard HP 20S manual *does* say that doing statistics destroys the number memories R_4 to R_9 , these being used to store the statistic values. Do users remember these restrictions or get caught out by them?

The Texas Instruments TI-30X Solar's \boxed{AC} clears all memories so it is impossible to clear the current calculation and retain the memory values. In contrast, the almost identical TI-30X's \boxed{AC} does not clear memories.

Suppose you wish to consign a result to memory on the Casio SL-807LU, which only provides $\boxed{M-}$ and $\boxed{M+}$, to subtract from and add to memory respectively. One solution — is it obvious to users? — is $\boxed{M+} \boxed{-} \boxed{MR} \boxed{M+}$ provided there is no overflow. The other solutions are to write the number down on paper or to hope that the memory contains zero. The purpose, if it is for calculation (rather than for marketing), of having a memory seems lost in such tedious or risky strategies.

Unless you are a perfect button-presser and have a perfect memory, you may not know what buttons have been pressed. On the HP 20S this is critical: a display of 1,234 could mean either 1.234 or 1234 depending on whether commas currently mean decimal points or thousands. The calculator doesn't tell you — *you* have to remember, or remember to do experiments to find out.¹

Again on the HP 20S, calculating large factorials, say $254!$, you get a wrong answer $1.E500$ (i.e., 10^{500}) — the calculator does display 'OFLO' before displaying this arbitrary number, but so briefly you would miss it if you were not watching closely.

Not all calculators give the same result for the same correctly entered calculation. The percent key illustrates this: it has no standard interpretation, and whatever it does on a particular calculator is generally inadequately defined. On a Casio *fx-82LB*, the sum $\boxed{1} \boxed{+} \boxed{5} \boxed{\%} \boxed{=}$ gives 120, yet a Casio SL-807LU gives 1.0526315. The Casio MS-270L gives 1.05263157894; so at least one calculator's rounding is incorrect.² The same sum on a Casio JW-8L or a Sharp EL-531GH give 1.05, more like "one plus 5% of one." (To get that result on the *fx-82LB* you would press $\boxed{1} \boxed{\times} \boxed{5} \boxed{\%} \boxed{+} \boxed{=}$.) See Figure 1.

Except the cheapest, most calculators promise to use normal algebraic notation. Most can work out $4 + 5 \times 6$ correctly (the MS-70L gets 54, not 34), but nevertheless have problems with more advanced operators.

On the TI-30X, to find how many combinations of 2 things can be taken from 5 (conventionally written $\binom{5}{2}$ or ${}_5C_2$) press $\boxed{5} \boxed{2nd} \boxed{x\%y} \boxed{2} \boxed{2nd} \boxed{nCr} \boxed{=}$. There is surely *no way* to work this out without the manual!³

$\boxed{x\%y}$, however, has another meaning: to swap sums around, so that $10 - 9$ can be 'swapped' to $9 - 10$. This overloading, using the same key for combinations and swaps, ensures that finding out a sum involving combinations cannot work. Even using brackets does not help work out $100 - \binom{5}{2}$:

¹This confusion could have been avoided by adopting the long-standing SI (Système International d'Unités) recommendation [3] to use either commas or dots between the integral and decimal parts of numbers, and, where digits are grouped in threes, *only* spaces.

²1.0526... is that number 5% less of which is 1.

³The two uses of the key $\boxed{2nd}$ itself are fairly clear. The \boxed{nCr} operation is an overloading of $\boxed{8}$, and therefore this key's second meaning is required. For clarity, explicit mention of $\boxed{2nd}$ or its equivalent is omitted in further examples.

Model	$ 4 \times - 5 ;$	$ 1 - 5 \% ;$	$ 1 + 5 \% $	Notes
Canon WS-121H	-1	-80†	1.05	† $ \bar{1} \bar{+} \bar{5} \bar{\pm} \bar{\%} $ calculates 0.95.
Casio MS-70L	-1	-80	1.0526315	
Casio MS-270L	-1	-80	1.05‡	‡1.0526315 displayed in a 2 decimal digit mode.
Casio JW-8L	-1	0.95	1.05	
Casio fx-85VH	-1	-80	120	
Casio fx-P401	-20	-80	120	
Sharp EL-531GH	-1	0.95	1.05	

Figure 1: **Examples of elementary calculations on various calculators.** Each calculation is preceded by $|\overline{AC}|$, and $|\overline{=}|$ is required to obtain answers except for percent calculations (exceptions are not shown).

Model	Sum: $\binom{5}{2}$
Hewlett-Packard 20S	$ \overline{5} \overline{INPUT} \overline{2} \overline{Cn, r} $
Canon F-800	$ \overline{5} \overline{a} \overline{2} \overline{b} \overline{nCr} $
Texas Instruments TI-30X	$ \overline{5} \overline{x\bar{y}} \overline{2} \overline{nCr} $
Sharp EL-531GH	$ \overline{5} \overline{nCr} \overline{2} $

Figure 2: **Variations in syntax for combinations.** Note the inconsistency in the F-800: the button is called nCr , yet the numbers are entered with a and b . (The table does not show the implicit use of shift keys.) See text for the obfuscation arising when using combinations in extended calculations.

pressing $|\overline{1}| | \overline{0} | | \overline{0} | | \overline{0} | | \overline{-} | | \overline{(|} | \overline{5} | | \overline{x\bar{y}} | | \overline{2} | | \overline{nCr} | | \overline{)}) | | \overline{=}|$ incorrectly obtains 10, not 90. (It appears to simply disregard the 100.)

Although $100 - \binom{5}{2}$ can be calculated by doing $\binom{5}{2} - 100$ followed by $|\overline{x\bar{y}}|$ (so it swaps the sum to the one intended), it appears $\binom{10}{2} \div \binom{5}{2}$ is impossible to express directly. One wonders how the TI-30X manual can claim it to have an

“algebraic operating system (AOSTM) [that] follows standard algebraic rules in completing calculations.”

The manual’s list of operator priorities immediately following this claim does not include the combinations key nor the key $|\overline{x\bar{y}}|$. Perhaps problems were to be expected.

The TI-30X is not alone with its problems for combinations: the HP 20S uses $|\overline{INPUT}|$ for combinations, and $|\overline{1}| | \overline{0} | | \overline{0} | | \overline{-} | | \overline{5} | | \overline{INPUT} | \dots$, as well as $|\overline{1}| | \overline{0} | | \overline{0} | | \overline{-} | | \overline{(|} | \overline{5} | | \overline{INPUT} | \dots | \overline{)})|$ both work out combinations taken from 95. Yet the Sharp EL-531GH works with a straight-forward $|\overline{5}| | \overline{nCr} | | \overline{2} |$, getting 10 without fuss, and the use of the key is consistent with its name, which suggests entering n then C (i.e., the key) then r . The $|\overline{nCr}|$ button on this calculator works as an ordinary infix operator, like $|\overline{+}|$ or $|\overline{\times}|$. It calculates $100 - \binom{5}{2}$ directly. There is no reason why combinations should not work so simply on other calculators. Figure 2 illustrates some approaches used to perform combinations.

The EL-531GH is not perfect, however, despite its “Direct Algebraic Logic,” which permits

“the easy entry of calculations as they appear, e.g., $\sin 30 + \cos 45$ is simply entered as $|\overline{\sin}| | \overline{3} | | \overline{0} | | \overline{+} | | \overline{\cos} | | \overline{4} | | \overline{5} | | \overline{=}|$. Calculations may be performed in an easy to understand manner from the simplest to the most complex equations.”

This suggests $2^{-\pi}$ might be worked out by $[\underline{2}][\underline{y^x}][\underline{-}][\underline{\pi}]$ or perhaps $[\underline{2}][\underline{y^x}][\underline{(-)}][\underline{-}][\underline{\pi}][\underline{)}]$. Neither approach works.

It is sensible to do some experiments, to see how a calculator works, and then, understanding it, proceed to the real problem. Once we can do the right sort of thing, we should be able to substitute the actual problem. So here we'll first find out how to do 2^{-2} , then try $2^{-\pi}$. This experiment will also give a feel for the sort of numerical answer to expect, so any mistakes should be easier to spot.

The obvious way of working out 2^{-2} : $[\underline{2}][\underline{y^x}][\underline{-}][\underline{2}][\underline{-}]$ gets zero, which is wrong; and $[\underline{2}][\underline{y^x}][\underline{(-)}][\underline{-}][\underline{2}][\underline{-}][\underline{)}]$ gets four, also wrong. Fortunately $[\underline{2}][\underline{y^x}][\underline{2}][\underline{+/-}][\underline{-}]$ gets 0.25 correctly. We then try $[\underline{2}][\underline{y^x}][\underline{\pi}][\underline{+/-}][\underline{-}]$ and get *no answer* except a little dash. So π and 2, both numbers, work in different ways! The fundamental strategy of trying sums you understand before doing the real problem does not work. The calculator promises an easy way of entering calculations, but it is flawed.

One solution is $[\underline{2}][\underline{y^x}][\underline{(-)}][\underline{0}][\underline{-}][\underline{\pi}][\underline{)}]$, which correctly gives 0.1133.... However, there is a symbol ‘ $\underline{\llcorner}(-)\llcorner$ ’ beneath $[\underline{+/-}]$ that looks promising. *The manual is silent on this feature.* Experiments suggest when used with a number (before, between or after digits) this function behaves as ‘change sign’ but otherwise it behaves as negate. Hence another solution is $[\underline{2}][\underline{y^x}][\underline{+/-}][\underline{\pi}]$, but the general success of $[\underline{2}][\underline{2}][\underline{+/-}]$, $[\underline{2}][\underline{+/-}][\underline{2}]$ and $[\underline{+/-}][\underline{2}][\underline{2}]$, which all result in -22 , suggest $[\underline{2}][\underline{y^x}][\underline{\pi}][\underline{+/-}]$ — which fails, with the calculator just showing two dashes. Although π is syntactically a number it is not implemented consistently; calculating with π is undefined and unpredictable.

This all seems to fall short of “Direct Algebraic Logic,” and the manual clarifies none of it.

The EL-531GH, *fx-P401* and many other calculators have fraction functions, $[\underline{a}\underline{b}\underline{c}]$ and $[\underline{d}\underline{c}]$ (usually on the same key) to enter and manipulate fractions. To enter $5\frac{1}{2}$, press $[\underline{5}][\underline{a}\underline{b}\underline{c}][\underline{1}][\underline{a}\underline{b}\underline{c}][\underline{2}]$. The result is displayed as $5\lceil 1\lceil 2$. Pressing $[\underline{d}\underline{c}]$ displays the result as a vulgar fraction, $11\lceil 2$, and *only then* $[\underline{a}\underline{b}\underline{c}]$ flips between $5\lceil 1\lceil 2$ and 5.5 alternately. However calculators cannot always convert a number displayed as 5.5 to $5\lceil 1\lceil 2$ (though there are standard algorithms to do so [9]): it appears that only numbers originally entered as fractions can be converted freely between decimal and fractional forms, and even such numbers soon ‘forget’ they are fractional. Pressing $[\underline{x^2}]$ is a case in point: a squared fraction is a fraction, but not on these calculators.

3 Systematic problems

The examples of obscurities perhaps disguise a more serious problem: calculators rarely report errors. This section now describes other problems common to many calculators. The arbitrariness exhibited in the previous section is bad, but there might have been a weak argument that the user should learn to cope with systematic problems — except that systematic problems are systematically avoidable.

3.1 Numerical problems

Calculating $1/6$ on almost any calculator obtains $0.16\dots 66$, not the correct $0.16\dots 67$. Rounding errors are ubiquitous (the HP 20S is an exception).

On the TI-30X, enter 111222333444, add 1, and you get 1112223335. The calculator has, without warning, ignored digits beyond those that fit in its display. Even if the manual had said that only ten digits were used for the entry of numbers, it should have taken the ten most significant digits. According to the manual, the calculator handles numbers provided the results are less than $9.999999999 \times 10^{99}$, and by calculating with “12 digits internally.” The test is within both these limits yet the answer is still wrong. Either the calculator should indicate an error or it should count the number of digits being pressed and get the right answer to the given precision (that this is possible is proved by doing it manually, using the $[\underline{EE}]$ button to enter the exponent 11).

The Microsoft Windows calculator, which emulates the ‘look and feel’ of a handheld calculator, gives 0.00 for $2.01 - 2$, gives 0.0000001000000002804 for $5.0000001 - 5$, and so on. It is hard to understand why these obvious numerical problems have remained uncorrected, even though the calculator has been

marketted for at least ten years and its serial number suggests it *has* been revised many times.⁴

Calculators have inadequate constants (recall the inconsistent use of π). The precision of conversions may be unnecessarily different to other calculations on the same calculator. All the calculators that provide the conversion, convert litres to US, not Imperial, gallons. This is not a problem if you want to convert US gallons to litres; it is a problem if you do not *know* that is what you are doing. Neither the calculators nor their manuals inform.

The *fx-P401* provides some SI prefixes (called ‘engineering symbols’ in its manual) but only eight, and ironically the most familiar: μ , m, k, M, p, n, G, T (selected in *that* peculiar order from two four-entry menus). Standard, but less familiar, prefixes such as exa and zetta — for which one might imagine greater need to look up, are not available.

3.2 Rearrangement

Regardless of implementation problems, conventional calculators (including RPN calculators) share a systemic problem: to perform any calculation, the sum has to be rearranged so that the answer (i.e., the press of the $\boxed{=}$ key) comes last. This is often difficult to do, and can result in additional problems for the user. The introduction gave the example $2^x = 100$, to solve for x ; and there are many others, such as calculating the diameter of a circle with area 17 square metre, equivalent to solving $\frac{\pi}{4}d^2 = 17$ for d .

For simple problems — and perhaps we have been conditioned into only attempting simple problems on calculators! — the rearrangement is routine. For other calculations the rearrangement is a major effort and will involve experimenting to check that the calculator’s detailed operation is sufficiently understood and is reliable. (Recall the problems of $\binom{5}{2}$ and $-\pi$ discussed in §2.) However, some seemingly simple problems, like $n! = 5040$, have no obvious solution.

Modern calculators could be considered a step backwards over slide rules, where $d = 3.15c$ (given c) and $c = d/3.15$ (given d) are the *same* calculation. The power of the slide rule, that many related calculations — including, in this case: $\sqrt{a} = 3.15c$; $\cos c = \frac{10}{3.15}\sqrt{1 - d^2}$; $c \tan 17.5 = \cos s$ and so on (Faber-Castell 62/82N; the a, c, d, s are the slide rule’s scale names) — can be read off at the same time, is exactly its main drawback: clutter and errors in reading the scales.

The Faber-Castell slide rule has one of the clearest manuals of all calculators discussed in this paper, and the only one that fits inside its pocket case. Also, it will withstand chemical and climactic conditions that would destroy its electronic usurpers.

3.3 Chain calculations

A chain calculation makes use of a previous result, contributing to the (users’ and calculators’) confusion over negate and subtract.

On all calculators examined here, -4×5 is a chain calculation that subtracts something (usually 20) from any previous result. Since the previous result is often zero, a user may not realise that negate, not $\boxed{-}$, has to be used to avoid this subtraction. Also, if -4×5 works by chance (because the previous result is 0) a user may expect 4×-5 to work.

Though the Casio *fx-P401* does 4×-5 correctly, -4×5 is always a chain calculation, and $\boxed{(-)}$ has to be used to avoid this. Yet it has \boxed{ANS} , for the value of the previous calculation, which obviates the need for default chain calculations, and hence the need for any special unary minus key. Although the *fx-P401* displays the entire calculation so a user is in principle aware of the chain calculation performed when it is not what they keyed, few other calculators do this.

Chain calculations complicate other functions. The *fx-115s* can display ‘9,’ when sometimes $\boxed{\sqrt{}}\boxed{=}$ gives 3, sometimes 0. The problem arises because the calculator does not distinguish whether $\boxed{=}$ has been pressed, and hence whether the 9 is a result (part of a chain) or whether 9 is the current number; moreover, it does not complain when there is an error. Calculators that do not aspire to so-called algebraic skills have no trouble, since roots are always suffix and work uniformly.

⁴Microsoft Windows NT Calculator version 3.50 (build 807) calculator is copyrighted 1985–1994.

3.4 Over-functionality

Buttons mean lots of different things. Buttons with four meanings, depending on the mode, are common. Many of the calculators (especially the Texas Instruments and Casio models) have low legibility and poor colour coding schemes, which exacerbate problems with overloaded buttons.

The corollary of overloaded buttons is that calculators provide two or three times as many functions as they have buttons; some even have thousands of functions (the Hewlett Packard HP 48G has “over 2 300”). A user may press a button and activate an unfamiliar feature, the consequences of which are difficult to sort out. Special functions typically have interactions with memories and other facilities: the only safe procedure after an error (if noticed!) is to start over.

School scientific calculators provide hyperbolic functions, yet do not provide the ordinary secant and cosecant. They provide tangent but not cotangent, despite both sine and cosine. This mixture of sophistication and omission is curious.

Worse is the support of complex numbers, which I discuss under the next heading.

3.5 Modes

What a calculator does, and what its buttons mean, depends on which mode it is in. The main modes are normal and statistics, but there are many more subtle modes, such as degrees/radians, or the HP 20S’s choice between commas and dots for the decimal point. Most calculators that provide degree/radians also provide grades, which are very easily confused with degrees (and, I think, of so little use the potential confusion is not worthwhile). Many calculators provide constant expressions, to change the behaviour (mode) of the calculator, e.g., so that $[\underline{2}][\underline{\times}][\underline{\times}]$ makes all subsequent calculations multiply by two (such features do not extend to functions such as percent or square root).

Complex mode (on all calculators reviewed here) unnecessarily stops certain functions — such as $[\underline{y^x}]$, $[\underline{\exp}]$, etc. — from working, and rarely are such errors reported. The *fx-115s* manual gives a list of functions that work unless in complex mode; mentioning functions like $[\underline{x^2}]$, which *does* work, but not functions like $[\underline{\%}]$, which *does not* work. Oddly, $[\underline{\sqrt}]$ does not work in complex mode, so $\sqrt{-1}$, that is i itself, cannot be calculated in complex mode! On the *fx-115s*, complex mode makes memories inaccessible: the overloaded key $[\underline{STO}]$ (otherwise store to memory) now exclusively means i (for the imaginary component of a complex number), its alternative meaning.

Some modes endure briefly, over one or two presses [16], e.g., $[\underline{2nd}]$ changes the meaning of the next key press to an alternative, also it changes $[\underline{2nd}]$ itself to get out of the mode. Where there are several mode buttons, they may interact in convoluted ways. The *fx-P401* has two shifts, $[\underline{\text{ALPHA}}]$ and $[\underline{\text{SHIFT}}]$ (as well as two mode buttons, $[\underline{\text{MENU}}]$ and $[\underline{\text{MODE}}]$). $[\underline{\text{ALPHA}}][\underline{\sin}]$ gets ‘D’ (either the register D or the hexadecimal digit D), but $[\underline{\text{SHIFT}}][\underline{\text{RCL}}][\underline{\text{ALPHA}}][\underline{\sin}]$, an attempt to store the value in register D instead gets the sin function, because the shifted recall (meaning store) also changes the mode to alpha, and $[\underline{\text{ALPHA}}]$ then gets out of *that* mode.

3.6 No undo, and feeble correction

To help correct mistakes, many calculators have a digit-correction button, $[\underline{\rightarrow}]$, though it only works with digits and no other buttons such as $[\underline{+}]$. On the TI-30X, $[\underline{\rightarrow}]$ doesn’t work with digits in an exponent, that is after $[\underline{EE}]$ has been pressed.⁵ The HP 20S has a correction key that corrects $[\underline{E}]$ as well (its version of the TI’s $[\underline{EE}]$), but why can’t it correct other mistakes, such as pressing $[\underline{C}]$? ... There is no reason why humans should restrict their mistakes to just the digit, decimal point and (only on some calculators) to $[\underline{E}]$, when there are others to choose from.

Most calculators take a sequence of operators to mean the last-pressed operator, to permit an incorrect operator to be corrected. This feature is another explanation for the 4×-5 problem, which

⁵ $[\underline{EE}], [\underline{E}]$ mean $n \times 10^m$ when pressed between suitable numerals n and m .

evidently means $4 - 5$. But if there is a ‘correction rule,’ it does not generalise: $[\underline{8}][\underline{3}][\underline{\sqrt{}}]$ takes a 6th. root, not a square root; and $[\underline{2}][\underline{+}][\underline{+}][\underline{3}]$ creates an ‘add 2’ mode (§3.5). Understood like this, operator correction is just another feature introducing spurious complexity and still failing to solve the original problem properly.

The *fx-P401* uses its display like a text editor so calculations can be edited keystroke-by-keystroke, using $[\underline{DEL}]$ to delete keystrokes. Yet $[\underline{DEL}]$ is just 3mm from $[\underline{AC}]$ (all clear), after pressing which a calculation cannot be recovered. There is no $[\underline{DEL}]$ for an accidental $[\underline{AC}]$.

3.7 Poor manuals

Calculator manuals are obscure (of the calculators discussed here, the Innovations Mathspad and HP 20S are notably better). Manuals give inadequate examples, from which it is impossible to work out exactly how to use them or what they are doing. Most Casio calculators’ manuals mix Spanish and English in alternate sentences. A calculation like $[\underline{1}][\underline{+}][\underline{5}][\underline{\%}]$ is not defined in any of the Casio calculators’ manuals examined, which may explain why it does different things on different models (Figure 1).

Calculators *require* manuals because they are so obscure to use, especially if you don’t use them routinely and therefore lack familiarity. It is surprising that no calculators have any provision for carrying the manual with the calculator, despite having cases that are almost, but not quite, big enough to do so. All calculators have a blank rear side, which could have been used for some helpful reference or reminders. Reminders would make a huge difference to how usable and reliable calculators are. The Mathspad is unique in having both a comprehensive manual (which also provides educational information) *and* a quick reference pamphlet that can be attached to the calculator’s wallet.

Simpler calculators have no manual, or just a ‘reference card’ printed on their bubble packaging. The MS-70L has its manual written on the inside of its box, which has to be cut open to be used. The EL-531GH has a tear-out page in the manual, but it is seriously deficient. The *fx-115s* has some notes on its lid, but these serve more to sell it than to help the user (who doesn’t need to be reminded it has 252 functions).

3.8 Hyperbole

The awkward aspects of calculators might be excusable if they were unavoidable, explained, or were unlikely to cause problems. One might not feel so critical of the manufacturers if they were honest about their gadgets’ capabilities. The EL-531GH is claimed to have 152 functions, but I don’t get a number anything as large as that, even including functions the manual does not mention!

The HP 20S has a “built-in program library including … complex number operations.” From this you might have expected something clearer than $[\underline{1}][\underline{INPUT}][\underline{2}][\underline{XEQ}][\underline{A}][\underline{3}][\underline{INPUT}][\underline{4}][\underline{R/S}]$ to work out $1+2i+3+4i$ (using A for +, B for –, and so on, despite having existing buttons to do addition and subtraction). Given that the calculator has keys $[\underline{y^x}]$ and $[\underline{e^x}]$, you might have thought it could do most things with complex numbers, but it cannot. Without the manual, this feature described prominently on its packaging is nearly useless. Like the Casio *fx-115s*, which also claims to be able to do complex operations, the HP 20S cannot find $\sqrt{-1}$.

The Mathspad manual (and its other promotional material) shows an equation longer than the display can handle;⁶ the pictures also give the impression that the actual 5×7 dot matrix display has a resolution around 10×14 .

Finally, if manufacturers want to call their calculators “visually perfect algebraic method” or “direct algebraic logic,” or whatever, they should find out how to do it ([1] is a standard text — dated 1985, earlier than any of these calculators).

4 Summary of problems, and a new approach

“… more a *belief* or *statement* which if expressed in English would run something like: *You type in an expression. The machine examines it, analyses it, and calculates the answer*

⁶The calculator is seemingly photographed as displaying $15\% \text{ of } £2.00 = 30p\sqrt{}$. In fact it can only do $15\% \text{ of } £2.00 = 30p\sqrt{}$, without a zero and without the spaces.

according to the rules of arithmetic." [24]

The purpose of a calculator is to do correct calculations, and to do so efficiently. It is clear that a calculator should relieve the user of the need to do mental operations and of the need to rely on paper, so far as possible. Calculators (and their manuals) should enable users to concentrate on what calculations they want solved, rather than having to experiment.

Most people, both manufacturers and users, assume calculators satisfy these requirements. Indeed modern calculators are so small, fast and convenient, one might have supposed they conform spectacularly well. This is, from the evidence raised above, a false impression. We saw examples of: bad parsing, obscure conventions, overloaded buttons, arbitrary restrictions, *ad hoc* fixes, no quick reference, inability to see the current calculation, and more. Users are deceived and disabled by horrible technology. To use a calculator one has to learn obscure and arbitrary incantations that have nothing to do with mathematics, nor have any compensating advantages, such as memorability, convenience or reliability.

I will now solve these problems.

The following discussion is closely based on a calculator I implemented for the Apple Macintosh computer; all the examples were copied from it. The Macintosh version differs from the description given here in some minor details by having, in particular: (i) a standard QWERTY alphabetic keyboard, so $\underline{\sin}$ is typed $\underline{s} \parallel \underline{i} \parallel \underline{n}$;⁷ (ii) a large display, upto screen size; (iii) user-definable colours and character styles for both user and calculator (the significance of this will become clear below); (iv) the ability to print calculations and save them in files; and (v) interactive reference and help material (reference summaries and 'balloon help'). The calculator is based on a primitive version that was described elsewhere [18], a reference that also discusses the underlying design principles and shows how they may be applied in other areas, not just to calculators.

The prototype uses commercial library functions for various operations, and some have numerical limitations that I did not correct — the prototype suffers from the sorts of problems mentioned in §3.1.

The prototype does not handle statistics, complex numbers, or certain higher functions (such as hyperbolic sines), though it could in principle. A Spanish user manual is not yet available.

5 The basic solution

My solution consists of the following main components:

- to take equations from the user, not instructions to calculate;
- to display exactly what the user has entered;
- to permit the equation to be edited;
- to fill in any missing numbers or symbols;
- to correct all mistakes, and ensure the result is numerically correct;
- and to do so at all times.

It is easiest to understand the new calculator by analogy with a word processor. A word processor has keys for entering text; the calculator has keys for entering equations, two arrow keys ($\underline{\leftarrow}$ and $\underline{\rightarrow}$) for moving the typing position left or right, and a delete key ($\underline{\text{DEL}}$) for deleting the symbol immediately to the left of the cursor. These facilities are sufficient for entering *any* equation, and for correcting any mistakes whatsoever. Also, the display shows exactly what has been entered, and nothing is concealed. Figure 8 shows what a basic model might look like. Note there need be no shift keys: each key has a single, fixed meaning.

So far the calculator may appear as an editing calculator, like the *fx-P401*. But its approach diverges: the calculator non-destructively completes the user's work, simultaneously correcting or solving any arithmetic mistakes or omissions. The calculator ensures everything is always numerically correct. How it does this is crucial to its success.

⁷As will become apparent, it is sufficient to type, say, \underline{i} , as the calculator itself completes any input.

(a) Copy and complete these number sentences.

$$7 \times \boxed{\quad} = 28$$

$$28 \div 7 = \boxed{\quad}$$

$$\boxed{\quad} \times 4 = 28$$

$$28 \div 4 = \boxed{\quad}$$

(b) Copy the following series of fractions and fill in the missing numbers.

$$\frac{3}{8} = \frac{40}{40} = \frac{120}{120} = \frac{200}{200} = \frac{21}{\boxed{\quad}} = \frac{60}{\boxed{\quad}} = \frac{375}{\boxed{\quad}}$$

Figure 3: **Primary school maths.** Typical examples, from (a) [11], (b) [10]. The new calculator solves these exercises directly, for example by keying (a) $\boxed{7} \boxed{\times} \boxed{2} \boxed{8}$, or (b) $\boxed{3} \boxed{\div} \boxed{8} \boxed{1} \boxed{4} \boxed{0} \boxed{\div} \boxed{1} \boxed{2} \boxed{0}$.

- A conventional calculator works out $4 + 5$ after the user *instructs* it with the presses $\boxed{4} \boxed{+} \boxed{5} \boxed{=}$. The new calculator requires an equation; as such, both ‘ $4 + 5$ ’ and ‘ $4 + 5 =$ ’ are strictly incomplete. The necessary completions would be ‘ $= 9$ ’ and ‘ 9 ’ respectively, and they are provided automatically by the calculator. The completions obtain the answer the user wanted, in fact the answer is available even before $=$ sign is pressed.
- If the user enters an invalid equation, such as $4 = 9$ this really needs correcting rather than merely solving! The calculator provides an ‘answer’ of $+5$, to balance the equation.

The notion of equations is new to calculators. Despite its advertising, the *fx-115s* cannot handle the equation $5 \times 2 + \sin 30 = 10.5$ because it needs instructions to evaluate an expression, $\boxed{5} \boxed{\times} \boxed{2} \boxed{+} \boxed{\sin} \boxed{3} \boxed{0} \boxed{=}$, and often far less obvious instructions for more complex calculations. In contrast, the new calculator directly accepts equations as they would be written. It is not ‘instructed,’ and the user never has to work out the ‘right’ way of instructing it to perform a calculation. Nor does the user press a key (such as $\boxed{=}$) to tell it when to display the answer; it always displays the answer, however much, or little, the user has keyed.

Note. In this article the calculator’s completions are distinguished thus: $\boxed{\text{completion}}$; this is the familiar convention from primary school (Figure 3a). The position of the editing cursor is shown using the symbol Δ . The actual calculator uses optional colours and text styles (blue, green, …; outline, bold, …) for completions, and uses a flashing vertical bar for the cursor.

The calculator shows at *every* stage of a calculation a completely correct display, *however it is edited*. If the calculator has just been picked up, and is doing no calculation, even this is completed: the calculator displays the equation $\boxed{0 = 0}$.

If the user enters the incomplete equation 4×-5 , the display shows $4 \times -5 \boxed{= -20}$. The user can type the $=$ sign themselves, anywhere, first or last, $4 \times -5 = \boxed{-20}$ or $\boxed{-20} = 4 \times -5$: thus $\boxed{=}$ is symmetric. Calculations can be chained using as many equals signs as required, as in $1 + \boxed{2} = 3 = 2 + \boxed{1}$; compare Figure 3b.

Examples already mentioned causing difficulties for conventional calculators are easily solved: entering $\boxed{2} \boxed{\uparrow} \boxed{1} \boxed{=} \boxed{1} \boxed{0} \boxed{0}$ (where $\boxed{\uparrow}$ is the new calculator’s equivalent to $\boxed{y^x}$) obtains a display of $2 \uparrow \boxed{6.6438562} = 100$; entering $\boxed{2} \boxed{\uparrow} \boxed{1} \boxed{=} \boxed{\pi}$ obtains a display of $2 \uparrow -\pi \boxed{= 0.11331473}$; entering $\boxed{!} \boxed{=} \boxed{5} \boxed{0} \boxed{4} \boxed{0}$ obtains $\boxed{7}! = 5040$. These preliminary examples, though demonstrating the ease of calculation, do not convey the style nor flexibility of interaction. (Examples below will show further how the calculator can be exploited very reliably and creatively by the user.)

Key press	Display after each key press
	$4 \times -5\Delta = -20$
$\underline{[←]}$	$4 \times -\Delta 5 = -20$
$\underline{[DEL]}$	$4 \times \Delta 5 = 20$
$\underline{[=]}$	$4 \times 1.25 = \Delta 5$
$\underline{[8]}$	$4 \times 21.25 = 8\Delta 5$
$\underline{[→]}$	$4 \times 21.25 = 85\Delta$
$\underline{[DEL]}$	$4 \times 2 = 8\Delta$
$\underline{[DEL]}$	$4 \times 1 = \Delta 4$

Figure 4: **Example of editing a simple calculation.** The changes to the display occur immediately, on every key press.

The calculator accepts *anything*. If the user enters the nonsense ‘.(/0’ the calculator displays $0.0 + (0 / (1 + 0)) = 0$. This is correct. If the user enters ‘4 = 5’ this is corrected and completed, and displayed as $4 + 1 = 5$. More interesting cases like $0 \uparrow 0$ (i.e., raising zero to the zeroth power, which is not mathematically defined) are also handled, in this case as $(1 + 0) \uparrow 0 = 1$. These examples look nasty as they are extreme cases — though it seems that users, at least initially, *enjoy* discovering such things!

Completions never alter the values of the user’s numbers. Although numerically valid, a completion for $0 \uparrow 0$ that is unacceptable for this reason is $0 \uparrow 0 = 1$.

How is entering calculations and editing them coordinated with completion? Assume the user has just entered 4×-5 and done no editing: the cursor will be positioned just after the 5, represented here by $4 \times -5\Delta$. Figure 4 gives examples of the user editing this calculation.

An important property is: *the same calculation always has the same completion.* The completion never depends on previous calculations nor on how it has been edited; it depends only on its actual text.

There are two more features:

There is a button $\underline{[FIX]}$, to ‘fix’ any correction in case it would be useful as part of a subsequent calculation. Suppose the display shows $1 = 1\Delta$, then pressing $\underline{[FIX]}$ changes it to $1 = 1\Delta$, *exactly* as if the user had themselves keyed the completion next to the Δ . The $\underline{[FIX]}$ key enables the user to take the answer of one calculation forward into another calculation.

Secondly, a handheld calculator’s display may be too small for some reasonable tasks. When the display size is insufficient, the calculator indicates that there is more calculation ‘off to the left’ or ‘off to the right’ but which cannot currently be displayed. Two arrows (\leftarrow and \rightarrow) in the display serve this purpose. To reveal more calculation in the direction of the arrow, the cursor is moved in that direction, and the display text scrolls to reveal the hidden parts of the calculation. (This feature is not needed on the prototype because of its large screen.)

6 Worked examples

“I took home the disc with your calculator on it, put it on our machine at home, and showed it to my 14 year old son. He said “That’s really neat,” but went off to play with the flight simulator. (He’s not very curious about computer things.)

Then, yesterday evening, he had a homework problem which involved using the Sine rule: $a / \sin(A) = b / \sin(B) = \dots$, in all kinds of different ways (i.e., to find a side, to find an angle). That meant turning things around, using Inv-Sine (postfix, of course!), and so on. Then he asked “Could we use that funny calculator you showed me?” We talked through how indeed

you can, and what's more you only need to input the formula once, thereafter you can just fill in the knowns and the calculator works out the unknown. So then he says: "WHY AREN'T ALL CALCULATORS BUILT LIKE THAT?" [25]

Showing a key-by-key description is tedious and obscures the clarity of this calculator. Consider, by way of comparison, a key-by-key description of comparable calculations on any other calculator!

6.1 Exploring Euclid's *Elements*

Euclid's *Elements* has many examples that can be tried on the calculator. Take Proposition 16 of Book V: "If four magnitudes be proportional, they will also be proportional alternately" [8].

Taking the numbers 3, 8, n , 40 from Figure 3b, we translate Euclid's proposition into modern notation and enter it into the calculator by typing $\boxed{3} \boxed{/} \boxed{8} \boxed{=} \boxed{/} \boxed{4} \boxed{/} \boxed{0}$; which gets $3/8 = \boxed{15}/40$ displayed (so the missing number is 15).

Then exchange the order of the numbers by editing. We see at once that $3/\boxed{15} = 8/40$. The 15 is the same as before; as Euclid said, the same magnitudes are also in proportion when taken alternately. We can also quickly change the 3 to a 35, 341, 341.8, ... and convince ourselves the proposition is general.

We will return to Euclid later.

6.2 Temperature conversion

Consider converting degrees Fahrenheit to Celsius. We will first assume that the formula cannot be recalled exactly, so we need to use the calculator to help find it. For the sake of brevity I will only explain one of the possible approaches — it must be emphasised that the way in which we work through the calculation is quite arbitrary. Unlike an ordinary calculator, there is no requirement to put calculations down in the 'right' order, or in one of a few acceptable orders. Any order is correct for the new calculator.

We know that the relation is linear, something like $F = C \times a + b$. The new calculator permits even such a skeletal equation to be entered: simply press $\boxed{=} \boxed{\times} \boxed{+}$. This would be displayed as $\boxed{1} = \boxed{1} \times \boxed{1} + \boxed{0}$. (For clarity, we drop the Δ if its position is not relevant.)

We know that $32F = 0C$; we can enter these numbers in the skeleton by moving the cursor left and right, and typing the numbers in the appropriate places. We obtain the display: $32 = 0 \times \boxed{0} + \boxed{32}$.

We will need to fix the calculated 32, which is done by moving the cursor to the correction and pressing \boxed{FIX} : $32 = 0 \times \boxed{0} + 32$.

Now recall that $212F = 100C$ (or some other relation, such as $-40F = -40C$): we enter these values in the appropriate places to determine the remaining unknown number. The display shows: $212 = 100 \times \boxed{1.8} + 32$. (Had the 32 value not been fixed, the calculation would have produced $212 = 100 \times \boxed{2.12} + \boxed{0}$ instead.)

Again, $\boxed{1.8}$ must be fixed, and we then have a skeleton that can be used for converting *any* Fahrenheit and Celsius temperatures. In Figure 5 we have deleted the 212 and 100, and are proceeding to find out what $-40F$ is.

7 Other features

The previous sections described what the calculator of Figure 8 could do. However, the prototype implementation is more comparable to a conventional scientific calculator in its capabilities.

7.1 'Scientific' functions

The prototype calculator adds additional functions: raising to a power $\boxed{\uparrow}$ (or more clearly, $\boxed{y^x}$), factorial $\boxed{!}$, and keys for $\boxed{\ln}$, $\boxed{\exp}$, $\boxed{\log}$, $\boxed{\sin}$, $\boxed{\cos}$ and $\boxed{\tan}$. A scientific notation button \boxed{e} is provided

Key press	Display after each key press
	$33.8 \Delta = 1 \times 1.8 + 32$
$\underline{-}$	$-\Delta 33.8 = 1 \times 1.8 + 32$
$\underline{4}$	$-4\Delta = -20 \times 1.8 + 32$
$\underline{0}$	$-40\Delta = -40 \times 1.8 + 32$

Figure 5: **Converting Fahrenheit and Celsius.** We continue with the calculation (described in the text) just after deleting the 100 and 212, so the cursor is positioned just to the left of the $=$ sign.

for conveniently entering very large and very small numbers. It is convenient but not essential to have buttons for inverse trigonometric functions: if you want the sine of 45, you enter $\sin 45$; if you want the arcsine of 0.5 you can enter $\sin = 0.5$.

When functions like logs and sines are used, there are opportunities for numerical problems. The log of a negative number is not defined; there is no sine whose value is greater than 1; factorials are defined only for non-negative integers. The calculator must correct any such errors, as, for example:

$$\begin{array}{ll} 5 = 5 \times \sin(90^\circ) & \parallel (0.95492966 \pi) ! = 6 \\ \log_{10}(1+0) = 0 & \parallel \log_{10}(-2) = 0.30103 \end{array}$$

The calculator provides a degree symbol ($^\circ$) in the sine calculation: this shows it is currently working in degrees and not radians. If it did not show the degree symbol there would be a hidden mode, which would be undesirable. *Always* the displayed calculation is correct, and completely correct; there is never any hidden information. The base of the logarithms (10, in these examples) is shown for the same reason.

7.2 Calculators with a large screen

A large screen, of several lines, supports more facilities. There are buttons to move up and down, $\underline{\uparrow}$ and $\underline{\downarrow}$, and the calculator begins to be more like a ‘mathematical word processor.’ The main new feature made possible is the table function, $\underline{\text{TABLE}}$.

The table key is simple and powerful. Pressing it makes ten copies of the current calculation, created on separate lines, but with a digit 0, 1, 2 … 9 inserted where the cursor was. Apart from its obvious use for generating tables, it can be used to systematise searching for particular values of functions — elementary examples being to find where lines cross axes.

A table of arcsines in $k\pi$ radians can be made by entering ‘ $\sin \pi = .$ ’ then moving the cursor to the right of the decimal point (to get, specifically, ‘ $\sin (0\pi) = 0.\Delta 0$ ’) and then pressing $\underline{\text{TABLE}}$:

$$\begin{aligned} \sin (0\pi) &= 0.\Delta 0 \quad \text{— initial equation} \\ \sin (0.03188428\pi) &= 0.1 \\ \sin (0.064094217\pi) &= 0.2 \\ \sin (0.096986684\pi) &= 0.3 \\ \sin (0.13098988\pi) &= 0.4 \\ \sin (0.16666667\pi) &= 0.5 \\ \sin (0.20483276\pi) &= 0.6 \\ \sin (0.24681669\pi) &= 0.7 \\ \sin (0.29516724\pi) &= 0.8 \\ \sin (0.35643371\pi) &= 0.9 \end{aligned} \quad \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \right\} \text{generated by } \underline{\text{TABLE}}$$

Problem	New calculator	Casio <i>fx-82LB</i> ‘fraction’ calculator
$4/3 = ?$	$4/3 = \boxed{1} + \boxed{1}/\boxed{3}$	$\boxed{4}/\boxed{3}$, answer: $1\frac{1}{3}$
$1.1 = ?$	$1.1 = \boxed{11}/\boxed{10} = \boxed{1} + \boxed{1}/\boxed{10}$	cannot handle decimals
$(4/3)^2 = ?$	$(4/3) \uparrow 2 = \boxed{16}/\boxed{9}$	cannot handle squares

Figure 6: Calculating with fractions.

The user can obtain more detail by repeating the table operation: e.g., moving to the 0.5 line, a nested table could be made to cover 0.50 to 0.59 (in steps of 0.01), or even 0.05 to 0.95 (creating a table from 0.00 to 0.95 in steps of 0.05).

7.3 Preferences for powerful calculations

For under-specified calculations, like ‘ $+ = 5$,’ there are many valid completions, including $\boxed{1} + \boxed{4} = 5$ and $\boxed{2.5} + \boxed{2.5} = 5$. The default behaviour of the calculator is to choose something ‘simple,’ in this case it adds zero (the $+$ operator’s identity), and the equation $\boxed{5} + \boxed{0} = 5$ is displayed.

The calculator provides many further options.

Rather than find identities, the calculator can be set to find equal parts. This provides simple solutions for addition such as: $10 = \boxed{5} + \boxed{5}$ and $10 \uparrow (\boxed{5} + \boxed{5}) = (10 \uparrow 5) \uparrow 2$, but, more interestingly, finds roots with multiplication: $10 = \boxed{2.1544347} \times \boxed{2.1544347} \times \boxed{2.1544347}$ and $16 = \boxed{4} \times (\boxed{2} \times \boxed{2})$.

Addition can be set to find powers of two: $10.5 = \boxed{0.5} + \boxed{2} + \boxed{8}$, and $2 \uparrow \boxed{0} + 2 \uparrow \boxed{4} = 17$.

Division can find reciprocals ($0.125 = \boxed{1}/\boxed{8}$), but is more interesting finding rational approximations: $\pi = \boxed{206921}/\boxed{65865}$ or $212 = \boxed{9}/\boxed{5} \times 100 + 32$.

Preferences can be used in combination with powerful effect. Addition set to ‘whole+part’ and division set to rational can do what no so-called fraction calculator can do: $0.75 + 2/3 = \boxed{17}/\boxed{12} = \boxed{1} + \boxed{5}/\boxed{12}$.

Multiplication can be set to find prime factors: $147 = \boxed{3} \times \boxed{7} \times \boxed{7}$. It can also be set to find powers of primes; if power is set to find (where possible) integer solutions, we can readily factorise. Thus, $28812 = \boxed{7} \uparrow \boxed{4} \times \boxed{3} \uparrow \boxed{1} \times \boxed{2} \uparrow \boxed{2}$. Likewise, it is possible to see that the rational approximation for π is in least terms: $\pi = \boxed{1447} \times \boxed{13} \times \boxed{11}/(\boxed{4391} \times \boxed{5} \times \boxed{3})$, as there are no common factors.

Although the prototype permits the preferences to be changed at will, feedback from users suggests that whole+part, rational division, prime factorisations and integer powers are very useful. A simplified calculator might therefore provide these settings permanently. The preferences work consistently well in combination; compare the natural and flexible behaviour of the new calculator with conventional calculators’ restrictive rules for handling fractions (Figure 6).

7.4 Memory registers

Conventional calculators have memory registers, and often different types of register, each with different conventions for their use. The new calculator has a simple memory feature.

Two buttons are required, called \boxed{STORE} and \boxed{RECALL} , or as many calculators would have it, \boxed{STO} and \boxed{RCL} .

The store key puts the number next to or containing Δ into the memory (there can only be one such number). The whole number is stored. The recall key inserts the stored number *exactly as if it had been entered by hand*, inserting it at the position of Δ .

An important property of \underline{RCL} is that whenever the memory is used its value can be seen (this is not the case on any other calculator reviewed here). The two immediately successive key presses \underline{STO} \underline{RCL} when the cursor is in a correction have a combined effect identical to \underline{FIX} , though also changing the memory.

This memory facility is identical to the prototype's, except the prototype follows the Macintosh user interface conventions [2]: the memory is called the 'clipboard' and the names of the operations are not store and recall, but copy and paste.

7.5 Constants and standard formulæ

A calculator is a convenient repository for tables of constants. There are two straight-forward ways they can be provided:

- As numerical values that are treated exactly like the user's own numbers (e.g., $2.99792458e8$ for the speed of light). This permits the constants to be edited if necessary, but can cause problems when a constant is entered 'in' a number: consider pressing $\underline{5} \underline{||} \underline{light}$ which gets $52.99792458e8$, probably erroneously.
- As symbolic constants (e.g., c for the speed of light). Pressing $\underline{5} \underline{||} \underline{light}$ would now more sensibly get $5c$ or $5 \underline{\times} c$ (depending on whether implicit multiplication is permitted).

The prototype calculator takes the first approach for a small database of constants and formulæ,⁸ and uses the second approach with implicit multiplication for π , currently its only symbolic constant.

The database makes the prototype calculator an enticing tool. Constants are selected from a group of hierarchical menus, under headings such as 'time' and 'distance.' All constants are in SI, the International System of Units [3]. A menu of SI prefixes covers all twenty, from yocto, 10^{-24} , to yotta, 10^{24} , and can be used to enter their value or as a convenient look-up table: the menu entries themselves state the terms, values and standard abbreviations.

Thus, to work out how many days in a year: select 'Julian year'; press $\underline{/}$; select 'Day' and it displays $31557600/86400 = 365.25$. Another question that can be easily answered directly is, "How far towards Alpha Centauri would 17cm tall 1 litre juice cartons stretch if piled on top of each other and filled with the Earth's oceans?" Other data is conventional recreational mathematics (drawing on [23] etc.), such as '2178,' which enters $4 \times 2178 = 8712$, showing that 4 times 2178 reverses its digits, or '2592,' which enters $2 \uparrow 5 \times 9 \uparrow 2 = 2592$.

7.6 Sound and errors

Although personal organisers and alarm clocks can make a noise, calculators are silent.

Sound can confirm that a key has indeed been pressed. (Typically users also like to silence the calculator, despite the additional security sound feedback provides that keys have indeed been pressed.)

Also, sound is extremely good at warning the user of errors. It may seem that the new calculator does not have 'errors' as such, but there are several situations where sound could be constructively used. The key $\underline{\Leftarrow}$ cannot sensibly do anything at the very left of the calculation, since it is not possible to move further left. It could beep a warning instead. The rule is: *beep if pressing a key does not cause any change to the display*.

The simple beep-if-nothing-happens rule appears to permit numerical overflow to be handled sensibly, but the requirements are not so simple as they might appear.

Consider an intended calculation of $1755!$ and assume it would be too large.

Which is preferable when $\underline{!}$ is pressed: to display $(0.9994302 \times 1755!)!_\Delta = 1.9792619e4930$,

⁸Constants are stored as strings, so it is easy to supply them with brackets, to avoid the problem mentioned earlier.

or to beep and not change the display?⁹ (The keyboard has a key $\lfloor e \rfloor$ so that large numbers can be entered exactly as they are displayed.)

The prototype calculator takes the approach that, if possible, no user action should be discarded. Thus we get the longer correction to huge factorials. The appropriateness of this rule is clear from examples such as trying 15 divided by 0.3. Until the digit 3 is entered, this forms a division by zero. It is inappropriate to ignore the zero, since this would confuse the user over how to enter 0.3. Instead the calculator shows $15/\lfloor 0 \lfloor +1 \rfloor = 15 \rfloor$, $15/\lfloor 0. \lfloor 0 + 1 \rfloor = 15 \rfloor$, then $15/0.3 \lfloor = 50 \rfloor$, for the intermediate steps, and the user has no problems. The user would be unlikely to be examining the display at each intermediate step of entering 0.3, so the changing corrections are not distracting.

7.7 Undo and editing

Although there are commercial calculators that permit the calculation to be edited, they suffer from poor editing. The editor on the *fx-P401*, unlike the proposed calculator, introduces two modes: *overwrite* and *insert*. Its transient insert mode is like the only (and permanent) method of editing on the new calculator, and the overwrite mode makes the calculator more complex without any gain. (The overwrite/insert tradeoff has been discussed at length elsewhere [5, 18]: the conclusion is that overwrite causes problems and cures none.)

To show that the calculator in Figure 8 does undo well enough it is sufficient to show that any key press can be undone simply.

There are four sorts of key:

- The ‘inserting’ keys — the normal keys that make the calculations, like $\lfloor 1 \rfloor \lfloor 2 \rfloor \lfloor + \rfloor \lfloor = \rfloor$ and so on — are undone by pressing $\lfloor \underline{DEL} \rfloor$, which deletes them.
- The ‘motion’ keys — the two keys $\lfloor \leftarrow \rfloor \lfloor \rightarrow \rfloor$ — move the cursor, and (presuming the cursor *did* move) are undone by pressing the other motion key.
- The $\lfloor \underline{FIX} \rfloor$ key converts a correction to something as if the user themselves keyed it in; this is undone by deleting the fixed stuff — this means that undoing this key may take one *or more* presses of $\lfloor \underline{DEL} \rfloor$.
- The $\lfloor \underline{DEL} \rfloor$ key itself. For this key, it seems the best form of correction is the user’s memory … re-press whatever has been deleted. This is straightforward, though it presumes the user knows what key the delete deleted.

A key $\lfloor \underline{UNDO} \rfloor$ used to undo accidental presses of $\lfloor \underline{DEL} \rfloor$ introduces more problems than it solves.

What should $\lfloor \underline{DEL} \rfloor \lfloor \underline{DEL} \rfloor \lfloor \underline{UNDO} \rfloor \lfloor \underline{UNDO} \rfloor$ do? Should it delete nothing, or should the second $\lfloor \underline{UNDO} \rfloor$ undo the first? Undo cannot in principle be expected to do everything [18].

Since the calculator deliberately avoids an ‘all clear’ button, it is not possible to delete so much that the potential complexity of $\lfloor \underline{UNDO} \rfloor$ is worthwhile.

7.8 Arbitrary limitations

A criticism of conventional calculators is that they have arbitrary limitations. The new calculator has three: the numerical range and precision, and the maximum size of calculation that can be handled. It works to 18 decimal digits within $\pm 1.2 \times 10^4$ ⁹³² with a maximum calculation of 32 000 characters: deliberately *far* larger than any normal problem that it would be used to tackle.

⁹The number 0.9994302 is $\frac{1754}{1755}$, easily be confirmed by setting division to rationals and entering $(\lfloor 1754 \rfloor / \lfloor 1755 \rfloor \times 1755)! = \lfloor 1754 \rfloor!$

7.9 Other important design details

Current calculators can be criticised on their non-technical design. A good calculator would be ruined by poor design. The following points may seem obvious when they are stated explicitly, but several calculators get details wrong.

The keys will have a firm response, so that there is definite tactile feedback when they have been pressed. Their legends will be clear (the new calculator has no modes, so only one label is required per key, and it can be written clearly in the middle of the key). The keys will not be highly reflective, to enable the legends to be read in strong lighting conditions.

If more than one key is pressed at a time, the calculator will do nothing (rather than something obscure).

The protective cover will be easily removable. It might be hinged onto the calculator so that it cannot be mislaid. The cover will provide space for the manual, or for its quick reference manual (if the calculator is sold with a substantial manual). *Also* the calculator will have a permanently attached quick reference table, perhaps on the inside of the cover, so that it can be read at the same time as the calculator is being used. This will provide useful advice, not empty hype to help sell it.

It is well known that demonstrations are helpful in explaining concepts; they are also helpful in selling gadgets (cf demo buttons on musical synthesisers). Moreover, a demo button can run self tests and report any diagnostics. (When you need the HP 20S's diagnostics, you won't be able to remember how to get them because they are invoked by unfamiliar, obscure and unlabelled key conventions.)

There will be a low-battery (and a low-light) warning that is displayed well before calculations become erratic or impossible.

The calculator will have a warranty and be supplied with addresses for correspondence in the country of purchase.

Obviously a customer gets what they pay for. A cheaper calculator cannot be engineered as well as a more expensive calculator. Nevertheless, the design effort of any calculator is essentially independent of its price since this cost is amortised over the lifetime of the product line. Calculators should be designed to the highest standards, and so that owners can rely on them to perform their intended task.

8 Possible objections to the new calculator

My solution may have problems that are only uncovered in large-scale use or by careful experiments; hence the need for this section that discusses objections to the new design. Each objection and my response begs empirical evidence that is not available.

8.1 The display changes too much

Here are several responses:

- The display on an ordinary calculator changes on almost every keypress. In fact, it could be said it is more confusing when it *doesn't* change.
- An argument by analogy to the way modern word processors behave is helpful. When text is being justified (i.e., maintaining straight margins) any correction to text, inserting or deleting a character, means the remainder of the paragraph has to be adjusted, and new page breaks appear or disappear. In the days before fast display updates, it was thought that continual justification would be a problem and various techniques were used to ameliorate it [15]. However, when displays work fast enough, it is little problem to the user, and the advantage of correct justification at all times (and a simplified user interface that does not need justified/not justified modes) is seen to be paramount. Likewise, the correctness of arithmetic at all times (and avoidance of before/after $\lfloor \rfloor$ modes) is paramount in the new calculator.
- Seeing the display change as numbers are changed, *but always being correct*, gives useful insight into calculations.
- If it is a problem, it can be ameliorated. Two technical suggestions have been made by many people, to ameliorate the supposed problem: the display need not change until the user pauses (this reduces display changes when the user is typing confidently); and the calculator could keep a log (narrative) of what the user is doing — this explains the nature of the changes.

$$\begin{array}{l} \log_{10}(3) + \log_{10}(4) = \log_{10}(12) \\ 71 \times \log_{10}(10) = \log_{10}(10) \uparrow [71] \end{array} \quad \parallel \quad \begin{array}{l} [10] \uparrow \log_{10}(42) = 42 \\ 2000 = [3] + \log_{10}(2) \end{array}$$

Figure 7: **The calculator as a ‘chalk board.’** The user can edit the equations and *see* the numerical identities, even when the base of the logarithm is changed. Note the effective use of preferences.

- I think that the problem of display changing is over-rated by the inevitably larger number of people who have *watched* (or read) demonstrations of the calculator, rather than actually used it. In practice¹⁰ the changing display is no problem for users and is not distracting.

8.2 The display is not big enough

This is a surprising criticism, when conventional calculators not only have a small display, which does not relate in any straight-forward way with the calculation, typically only showing the last number entered or calculated and some simple status information.

8.3 The new calculator does not have a familiar user’s model

This criticism would have applied to any calculator when it was first introduced. Given the earlier discussion in this paper, it is not clear what the advantage of a ‘familiar’ user’s model is supposed to be: it would either be very complicated or wrong, and usually both complicated and wrong.

David Pullinger has suggested that the new calculator is like a chalk board: you write down things on the board, making corrections as you go, and the calculator non-intrusively completes everything. With a conventional chalk board, you need two hands, one to write with and one to rub out with. The new calculator spares you that, and takes on the numerical chores (Figure 7).

8.4 Children should learn how to break calculations down into small steps

To use a conventional calculator certainly requires a problem to be broken down into simple steps, otherwise operators will conflict and the results will be unreliable. It is not clear to me that if a calculator purports to handle composite calculations that doing it wrongly is the way to help children learn. The new calculator handles large equations naturally, and there is no need (though it is possible) to break equations into smaller steps. The calculator is analogous to a spelling checker: it doesn’t so much help you learn to spell, as help you get on and do writing.

8.5 It is misleading to provide answers to incorrect questions

This might be a valid criticism if the calculator took questions and *then* provided answers. Instead, the calculator *cooperates* with the user and together they find a correct equation. It is hard to appreciate this point merely by reading a paper (with static examples), and without being involved in interacting with the calculator for oneself. More generally, it is limiting to suppose it necessary that people use computers (or calculators) in a narrowly defined ‘correct’ way, when permissive machines [21] can easily cooperate with and adapt to their users, as the new calculator can so easily.

If you entered $[\underline{4}][\underline{+}][\underline{=}][\underline{9}]$ to an ordinary calculator, it would certainly be wrong for it to provide any answer other than an error report. It is not possible to *ask* that of the new calculator. Before hitting $[\underline{=}]$, the calculator is already showing $4 + [0 = 4]$; by the time you have together got to the $[\underline{9}]$, the calculator has offered $\boxed{5}$.

What of ‘answers’ to $\pi!$ or to $1/0$? Again, the user does not ask questions. It is more like being in a conversation with a helpful teacher who says, “But *we* can say $1/(0 + 1) = 1$.” Very little practice is required to have powerful and useful conversations.

¹⁰Based on a 20cm computer display, not a hand-held (vertically oriented) calculator’s $\approx 6\text{cm}$ display.

8.6 Non-specific complaints

My final response to objections is that the new design is mathematical. It is mathematical in two important ways that no other calculator is. It is declarative. It is general.

Mathematics itself is declarative, meaning its notation is used to express (declare) facts whose meaning do not depend on how, when or where those facts are expressed. This ensures that mathematical operations, such as substituting an expression with its value, are sound. Otherwise the value of an expression might change depending on how it was expressed, so replacing with a supposedly equal value could change it. A declarative calculator has a more natural match with mathematics than one that is not. It means that users can concentrate on *what* they want to calculate, rather than on *how* to rearrange and express their calculation into the arcane and arbitrary steps the calculator understands and will perform correctly.

Mathematics can have no success where it cannot generalise. To paraphrase Peirce [14]: one cannot deny that conventional calculators are mathematical, after a fashion; but, owing to the exceptions that everywhere confront the user — such as the different behaviour of constants and numbers; arbitrary conventions for certain operators; undefined functions — there results a calculator whose wings are effectually clipped, and which can only run along the ground. Not so the new calculator.

9 Possible developments

The new calculator could be extended in many ways. It could also be simplified: a version with just the four arithmetic operations would be very clear and still powerful. Such a simple, powerful and consistent calculator would be widely appreciated.

As far as the user is concerned, overflow occurs arbitrarily and has strange effects, such as the correction to large factorials. An explicit symbol might be preferable,¹¹ and indeed one that the user could enter. Division by zero could be handled as a fraction unexceptionably as $1/0 = \boxed{1/0}$.

If required, a % key could be introduced, and it could very easily be defined so that the calculator corrected any use to a canonical calculation, where %'s meaning would be lucid. My view, however, is that percentage is not a function for a calculator, rather a convention for talking about ratios that has to be learnt, just as telling the time or knowing that when you are twelve you are in your thirteenth year; these require clear thinking about numbers, not a calculator.

It is straight forward to introduce an explicit template system, so a collection of standard (perhaps annotated, perhaps user-extensible) skeleton formulae is made available. This would be useful in statistics, especially for significance tests to be used 'both ways.' In particular, a skeleton could be fixed despite arbitrary edits to its variable slots, hence guaranteeing the formula's correctness. A template system could be integrated with a demonstration mechanism (§7.9).

A calculator could learn what calculations it is used for. This is a generalisation of the previous suggestion; Bell, Cleary and Witten [4] show this may be done effectively for a conventional calculator.

The table feature assumes a large (a multi-line) display. Such a display (if dot matrix) could also provide a graph of a calculation's values over a given range.

If you know what a table is about, the current approach of repeating the formula on every line is excessive (though it preserves the declarative rule that every line is a correct calculation). Instead, the table could be abbreviated to show only the varying numbers; it could be typeset with column headings and rules. Another possibility (taking an idea from Mathspad) is to use the table facility to generate questions for the user.

Most conventional calculators provide various display formats for numbers. An elementary calculator should work in decimals; a scientific calculator in scientific notation, perhaps with an option to make the power of ten a multiple of 3 (so numbers are displayed in SI multiples); and a business calculator should work in exact numbers to the given precision. Unfortunately if the decimal precision is zero, the user could expect integers, but integer (Diophantine) problems are hard — consider entering $\uparrow 2 + 2 = \uparrow 3$ expecting $\boxed{5} \uparrow 2 + 2 = \boxed{3} \uparrow 3$. Thus confusion can be expected if the user changes from one format to another, especially if precision is lost. Conclusion: *A variety of formats is a bad idea; the calculator should be specialised to its intended area of use.*

¹¹ \perp , not ∞ , is the conventional symbol for an undefined quantity. The prototype uses ?, but unfortunately the user cannot enter it.

It may be desirable to introduce modes for working in degrees, changing the logarithm base, or setting other preferences. This can be done consistently with the rest of the design, though the prototype uses menus for this purpose. We might introduce a button $\underline{\underline{\wedge}}$ and a status (mode) line in the display. The new key moves up to the status line and down from it. The status line is edited in *exactly* the same way as the ordinary calculation display; it shows the Δ symbol, moved by $\underline{\leftarrow}$ and $\underline{\rightarrow}$ in the usual way. The difference is that the syntax the status line interprets is not arithmetic but mode information.

One way to implement this is to give keys secondary labels ('Degrees,' 'Radians,' etc) which are the keys' meanings when the cursor is in the mode line. There are other approaches, but we do not have space to discuss the design tradeoffs.

9.1 Going beyond the handheld paradigm

Mathematical calculations are conventionally two dimensional, but the calculator uses a one dimensional notation, which, in particular, makes fractions unnecessarily hard to read. Two dimensional equation editing is well-established [7], and can be done in a fashion that is compatible with the calculator's design requirements. Two dimensional editing is also be appropriate for elementary calculations [17]:

$$\begin{array}{r} 17 \\ 23 \\ \hline 40 \end{array}$$

Calculators are embedded in other products, such as watches, personal organisers, 'personal digital assistants,' and computer workstations. Proper integration with the whole product is a crucial part of the design. In all cases the calculator is a separate concept, but numbers aren't to the user.¹² Everywhere a number is permitted, a calculation should be permitted. This would also makes systems more useful: anywhere a number is required, the user can enter an expression for it. The number defined is the numerical correction to the user's calculation.

The rule is graceful: if the user enters a simple expression, say 6 or 2×3 , the correction is its value. If the user enters an equation such as $12/ = 2$, the embedded correction (here $\boxed{6}$) defines the value.

9.2 Names

The calculator deliberately avoids names, making it simpler to use, but some users expect it to be able to solve more sophisticated problems. The simplest extension is a preference (§7.3) that requires all unknowns to be equal. This enables solutions of polynomials to be found: e.g., $\boxed{3} \uparrow 2 + \boxed{3} = 12$, rather than the current $\boxed{3.4641016} \uparrow 2 + \boxed{0} = 12$.

The calculator cannot 'talk about' its own calculations. A function $u(n)$ could be the value of the calculation on line n , and if $\#$ denotes the current line number, the value from the previous line would be written $u(\# - 1)$. This is very flexible; e.g., a table based on $u(\# - 1) + 1/\#!$ generates a series that converges to e . Unfortunately permitting expressions like $u(u(\#))$ requires constraint analysis and a conflict resolution strategy.

All schemes for introducing names certainly enhance the power of the calculator, but at the cost of increased complexity, conceptually for the user as well as for the implementor. The current design assumes the calculator is fast, but any adequate notation for functions expresses functions that need not terminate or are indefinitely slow. The simplest extension making unknowns equal fails even for $+ 1 =$.

It is not clear that a satisfactory scheme can be devised that does not engender various exceptions or limitations in the calculator's capabilities. Such complexities are not comprehensible in elementary mathematics, the intended domain of the calculator, and therefore should not be adopted. Moreover, names enable a different and more powerful style of problem solving (see Polya's excellent text [13]): a calculator that made it natural to confuse abstract names and concrete numbers, that gave no insights

¹²The Casio SF-4300 is an example: it has two $\underline{\underline{\%}}$ keys, one which works as text, the other works with the calculator.

The main $\underline{\underline{\%}}$ key enters 5 when working as a calculator!

into algebra rather than numerical coincidences, would cause more educational damage than the calculators we are trying to supercede. (That wouldn't stop it selling.) Compare our brief exploration of Euclid (§6.1) with Euclid's own exposition of 300BC. Euclid proves a proposition is true for all magnitudes; we only demonstrated a few specific cases. We have made progress in calculating but not in thinking.

10 Conclusions

This article exhibited severe and widespread problems with conventional calculators. It then provided a new approach, notable for its elegance: more powerful and more flexible than conventional calculators, yet more restrained. There is every reason to suppose it is much easier to use. It has fewer bugs in its implementation, and it has fewer problems in its use, even when employed to perform sophisticated calculations that would be beyond the safe range of conventional calculators. The new calculator was described in detail, both to define it and also to emphasise the coherence of its design. The calculator is easy to implement in the same technology as existing calculators.¹³

We may surmise that the inevitably poor fit between the imperative style of conventional calculators and their intended use for calculation, whose semantics are declarative, results in unavoidably poor design and low usability. Poor quality control arises because imperative designs are easy to extend without regard to design coherence, simplicity, ease of use, reliability or any other desirable characteristics. (However this mismatch fails to explain the technical incompetence of many features of the designs reviewed in this paper.) Once a button is put on a calculator it is easy to make it do things without any consideration for whether these things are helpful or relate coherently with other functions. The TI-30X's bizarre $\lfloor \overline{x} \overline{y} \rfloor$ key is a case in point.

An advantage of a declarative design is that no extension can be considered in isolation; every putative extension has technical implications everywhere, just as it has usability implications everywhere. For a declarative application, such as calculators, we claim technical issues and usability issues can coincide. The new calculator is good. Any proper extension will maintain its quality.

References

- [1] A. V. Aho, R. Sethi and J. D. Ullman (1985) *Compilers: Principles, Techniques and Tools*, Addison-Wesley.
- [2] Apple Computer Inc. (1992) *Macintosh Human Interface Guidelines*, Addison-Wesley.
- [3] R. J. Bell, UK editor (6th. ed, 1993) *SI: The International System of Units*, Approved translation of the International Bureau of Weights and Measures, *Le Système International d'Unités (SI)*, HMSO.
- [4] T. C. Bell, J. G. Cleary, and I. H. Witten (1990) *Text Compression*, Prentice Hall.
- [5] R. Bornat and H. W. Thimbleby (1989) "The Life and Times of Ded, Display Editor," *Cognitive Ergonomics and Human Computer Interaction*, pp225–255, J. B. Long and A. Whitefield (eds.), Cambridge University Press.
- [6] Casio Electronics Company Ltd. (1994) "For Complicated Calculations. Uncomplicated Calculators." *The Times*, Sept. 7, p13.
- [7] Design Science Inc. (1992) *MathType: Mathematical Equation Editor User Manual*, Long Beach, CA.
- [8] Euclid (ca 300BC) *The Elements*, , translated by T. L. Heath (1990) Encyclopædia Britannica, *Great Books of the Western World*, **10**, 2nd. ed.
- [9] G. H. Hardy and E. M. Wright (5th. ed., 1979) *An Introduction to the Theory of Numbers*, Oxford University Press.
- [10] B. V. Hony (1992) *Active Mathematics 1*, Longman.

¹³A custom chip is currently being developed by students at Middlesex University.

- [11] P. Latham and P. Truelove (1983) *Nuffield Maths 3, Pupil's Book*, Longman.
- [12] R. E. Mayer and P. Bayman (1981) "Psychology of Calculator Languages: A Framework for Describing Users' Knowledge," *Communications of the ACM*, **24**(8), pp511-520.
- [13] G. Polya (Combined Edition, 1981) *Mathematical Discovery*, John Wiley & Sons.
- [14] C. S. Peirce (1960) "The Simplest Mathematics," *Collected Papers of Charles Sanders Peirce*, **IV**, ¶236, p196, C. Hartshorne and P. Weiss (eds.), Harvard University Press, 1960.
- [15] H. W. Thimbleby (1983) "Guidelines for 'Manipulative' Editing," *Behaviour and Information Technology*, **2**(2), pp127-161.
- [16] H. W. Thimbleby (1982) "Character Level Ambiguity: Consequences for User Interface Design," *International Journal of Man-Machine Studies*, **16**, pp211-225.
- [17] H. W. Thimbleby (1986) "The Design of Two Innovative User Interfaces," *Proceedings British Computer Society Conference on Human Computer Interaction*, HCI'86, M. D. Harrison and A. F. Monk (eds.), pp336-351, Cambridge University Press.
- [18] H. W. Thimbleby (1990) *User Interface Design*, Addison-Wesley.
- [19] H. W. Thimbleby (1992) "The Frustrations of a Pushbutton World," in *1993 Encyclopædia Britannica Yearbook of Science and the Future*, pp202-219, Encyclopædia Britannica Inc.
- [20] H. W. Thimbleby (1993) "Computer Literacy and Usability Standards?" in *User Needs in Information Technology Standards*, pp223-230, C. D. Evans, B. L. Meek and R. S. Walker (eds.), Butterworth-Heinemann.
- [21] H. W. Thimbleby (1994) "Technological Hubris," *Computers & Digital Techniques*, **5**(5), pp229-234.
- [22] L. R. Weiner (1993) *Digital Woes*, Addison-Wesley.
- [23] D. Wells (1986) *The Penguin Dictionary of Curious and Interesting Numbers*, Penguin.
- [24] R. M. Young (1981) "The machine inside the machine: users' models of pocket calculators," *International Journal of Man-Machine Studies*, **15**, pp51-85.
- [25] R. M. Young (1994) *personal communication*.

Acknowledgements

Richard Young made extensive and pertinent comments on a draft of this paper. Ann Blandford, Alan Dix, Dick Gledhill, Thomas Green, David Pullinger, Colin Runciman and others have helped considerably with clarifying the rationale underlying the new calculator. The referees made substantial and constructive comments.

For an executable version of the calculator described in this paper, requiring a Macintosh running System 7, send £20, including p&p, to Prof. Harold Thimbleby (Calculator), Department of Computing Science, Middlesex University, Bounds Green Road, London, N11 2NQ. Cheques should be made payable to Middlesex University.

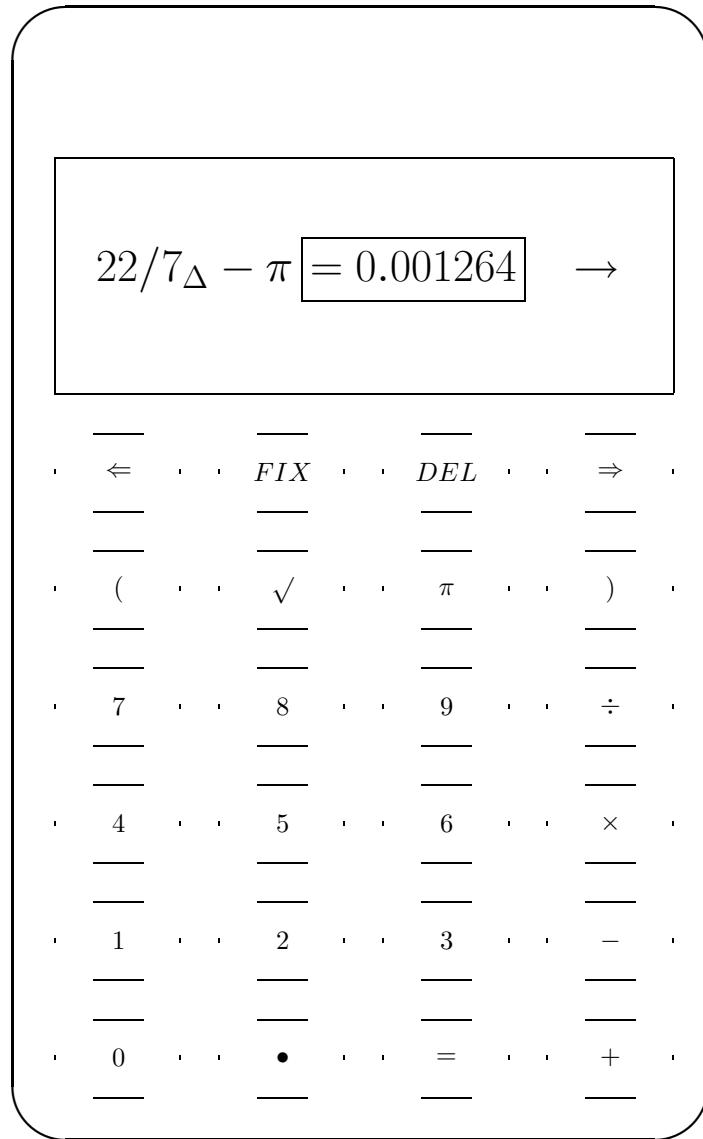


Figure 8: **The new calculator — example key layout.** The calculator has a protective flip cover with a quick reference list, explaining, in particular, the buttons \boxed{FIX} , \boxed{DEL} , $\boxed{\leftarrow}$, $\boxed{\rightarrow}$, the display's Δ symbol (which is moved by the arrow keys) and the significance of the symbols \rightarrow and \leftarrow that may appear in the display, which indicate more information off to the left or right of the display. In the display shown, there is more to the calculation to the right of the display; more expensive models would have a larger display, perhaps extending to several lines. This schematic diagram does not show key colours, nor the hidden photocell panel that ensures the calculator is 'on' whenever lighting conditions are anyway sufficient for it to be used. An internal backup battery (rechargeable from the photocells) ensures the calculation is not lost when the calculator is in darkness (or turned over).